CSC 461 HW#1, Kefu Zhu

Problem 1

(a) Insert < 'Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE.

Answer: This operation should be rejected because of violation of **domain constraint** and **inherent constraint**. However, if we modify the value of Bdate to correct DATE data type and modify the value of Sex to correct CHAR data type. This operation will be valid

- **Violation of domain constraint**: The value of **Bdate** attribute in the new tuple ('21-JUN-42') has **TEXT** data type rather than the required **DATE** data type
- **Violation of implicit constraint**: The value of **Sex** attribute in the new tuple (M) has invalid data type rather than the required **CHAR** data type

(b) Insert < 'ProductA', 4, 'Bellaire', 2 > into PROJECT.

Answer: This operation should be rejected because of violation of **referential integrity**. However, if new tuple with a value of 2 for Dnumber attribute is added to the Department relation first (A new department is defined in the Department relation with Dnumber of 2). This operation will be valid

- **Violation of referential integrity**: The value of **Dnum** attribute in the new tuple (2) does not correspond to any existing primary key value in the referenced relation

(c) Insert < 'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.

Answer: This operation should be rejected because of violation of **key constraint**, **referential integrity** and **domain constraint**.

- **Violation of key constraint**: The value of **Dnumber** attribute in the new tuple (4) already exists in the relation
- **Violation of referential integrity**: The value of **Mgr_ssn** attribute in the new tuple ('943775543') does not correspond to any existing primary key value in the referenced relation
- **Violation of domain constraint**: The value of **Mgr_start_date** attribute in the new tuple ('01-OCT-88') has **TEXT** data type rather than the required **DATE** data type

(d) Insert < '677678989', null, '40.0' > into WORKS_ON.

Answer: This operation should be rejected because of violation of **referential integrity**

- **Violation of referential integrity**: The value of **Essn** attribute in the new tuple ('677678989') does not correspond to any existing primary key value in the referenced relation

(e) Insert < '453453453', 'John', M, '12-DEC-60', 'SPOUSE' > into DEPENDENT.

Answer: This operation should be rejected because of violation of two **domain constraints**. However, this operation would become valid if the data type issues listed below are corrected

- The value of **Sex** attribute in the new tuple (M) has invalid data type rather than the required **CHAR** data type
- The value of **Bdate** attribute in the new tuple ('12-DEC-60') has **TEXT** data type rather than the required **DATE** data type

(f) Delete the WORKS_ON tuples with ESSN= '333445555'.

Answer: This operation is valid

(g) Delete the EMPLOYEE tuple with SSN= '987654321'.

Answer: This operation should be rejected because of violation of **referential integrity??????**. However, this operation will be valid if all tuples in other relations that have foreign key referencing the '987654321' of **SSN** are removed or modified such that they are referencing other existing values, before this DELETE operation.

- The **Super_ssn** ('987654321') of the third tuple in **EMPLOYEE** relation does not correspond to any existing primary key value in the referenced relation
- The **Mgr_ssn** ('987654321') of the 2nd tuple in **DEPARTMENT** relation does not correspond to any existing primary key value in the referenced relation
- The **Essn** ('987654321') of the 14th and 15th tuples in **WORKS_ON** relation do not correspond to any existing primary key value in the referenced relation
- The **Essn** ('987654321') of the 4th tuple in **DEPENDENT** relation does not correspond to any existing primary key value in the referenced relation

(h) Delete the PROJECT tuple with PNAME= 'ProductX'.

Answer: This operation should be rejected because of violation of **referential integrity???????**. However, this operation will be valid if all tuples in **WORKS_ON** that have **Pno** value of 1 are removed or modified such that the new values can be referenced to existing values in the **Pnumber** attribute in **PROJECT**, before this DELETE OPERATION

- The **Pno** (1) of the 1st and 4th tuples in **WORKS_ON** relation does not correspond to any existing primary key value in the referenced relation

(i) Modify the MGRSSN and MGRSTARTDATE of the DEPARTMENT tuple with DNUMBER=5 to '123456789' and '01-OCT-88', respectively.

Answer: This operation should be rejected because of violation of **domain constraint**. However, this operation will be valid if the data type of the new value of **Mgr_start_date** is corrected to the required DATE data type

- The value of **Mgr_start_date** attribute in the new tuple ('01-OCT-88') has **TEXT** data type rather than the required **DATE** data type

(j) Modify the SUPERSSN attribute of the EMPLOYEE tuple with SSN= '999887777' to '943775543'.

Answer: This operation should be rejected because of violation of **referential integrity**. However, this operation will be valid if, before this operation, an employee with SSN value of '943775543' is added into the EMPLOYEE relation

- The value of **Super_SSN** attribute in the new tuple ('943775543') does not correspond to any existing primary key value in the referenced relation

(k) Modify the HOURS attribute of the WORKS_ON tuple with ESSN= '999887777' and PNO= 10 to '5.0'.

Answer: This operation is perfectly valid

Problem 2

(a) Give the operations for this update

The operation on **AIRPORT**,**AIRPLANE_TYPE** and **CAN_LAND** should be performed once (enter information of all known airplane types from all major companies, and the landing permission for different types of airplanes for all major airports) and updated on a regular basis. The operation below does not depend on anything else

Operation on AIRPORT: INSERT <'ORD','Chicago O'Hare International Airport','Chicago','Illinois'>;

Operation on AIRPLANE_TYPE: INSERT <'KZ747', 31, 'Unicorn Airline'>;

Operation on CAN_LAND: INSERT <'KZ747','ORD'>; INSERT <'KZ747','ROC'>;

A valid sequence of operations to complete a new update in the **AIRLINE** database should looks like the following, in roughly the same order:

1. **Operation on FLIGHT relation**: INSERT <'AA12345','American Airline',0> INTO FLIGHT;
2. **Operation on FARE relation**: INSERT <'AA12345', 'wow', 1000, NULL>;
3. **Operation on FLIGHT_LEG relation**: INSERT <'AA12345', 1, 'abc','2018-01-01','xyz','2018-01-02'> INTO

FLIGHT_LEG;

4. **Operation on LEG_INSTANCE relation**: INSERT <'AA12345', 1, '2018-01-01', 30, 'AAIDK', 'ORD', '2018-01-01 23:10:00', NULL, NULL> INTO LEG_INSTANCE;
5. **Operation on LEG_INSTANCE relation**: INSERT <'AA12345', 1, '2018-01-02', 30, 'AAIDK', NULL, NULL, 'ROC', '2018-01-02 06:00:00'> INTO LEG_INSTANCE;
6. **Operation on SEAT_RESERVATION relation**: INSERT <'AA12345', 1, '2017-12-10', 'B4', 'Kefu Zhu', '2178986825'>;
7. **Operation on AIRPLANE**: INSERT <'AAIDK',31,'KZ747'>

(b) What types of constraints would you expect to check?

I expect to check domain constraints, application based constraints, key constraints, entity integrity constraints and referential integrity constraints

Below are some examples for domain constratins and application based constraints:

1. All departure time should be before its corresponding arrival time (e.g. FLIGHT_LGE.Scheduled_departure_time < FLIGHT_LEG.Scheduled_arrival_time)
2. If we are using this AIRLINE database for entering all data before any actual arrangements, then all new data records should not have DATE values that are in the past (e.g. FLIGHT_LEG.Scheduled_departure_time < CURDATE() **or** GETDATE() **or** NOW())
3. The result of converting the Scheduled_departure_time in a flight's first FLIGHT_LEG should be the same as the value of Weekdays in the FLIGHT relation (e.g. WEEKDAY(FLIGHT_LEG.Scheduled_departure_time WHERE FLIGHT_LEG.Leg_number = 1) = FLIGHT.Weekdays)
4. The SEAT_RESERVATION.Customer_phone should be in valid format
5. The type of airplane should be acceptable at both the departure and arrival airports
6. AIRPLANE.Total_number_of_seats $\leq$ AIRPLANE_TYPE.Max_seats for the same type of airplane

(c) Which of these constraints are key, entity integrity, and referential integrity constraints and which are not?

- **Key constraint**: The values in **Airport_code** attribute of **AIRPORT** relation has duplicated values
- **Entity integrity constraint**: The values in **Flight_number** attribute of **FLIGHT** relation has NULL values
- **Referential integrity constraint**: The values in **Arrival*airport*code** attribute of **FLIGHT_LEG** relation contain value that does not exist in **Airport_code** attribute of **AIRPORT** relation

The rest constraints are not key, entity integrity and referential integrity constraints

(d) Specify all the referential integrity constraints on the figure

Define: A → B as A reference to B, where A is foreign key and B is primary key

1. FLIGHT_LEG.Flight_number → FLIGHT.Flight_number
2. FLIGHT_LEG.Departure_airport_code → AIRPORT.Airport_code;
3. FLIGHT_LEG.Arrival_airport_code → AIRPORT.Airport_code;
4. FARE.Flight_number → FLIGHT.Flight_number
5. LEG_INSTANCE.Flight_number → FLIGHT.Flight_number
6. LEG_INSTANCE.Leg_number → FLIGHT_LEG.Leg_number
7. LEG_INSTANCE.Departure_airport_code → AIRPORT.Airport_code;
8. LEG_INSTANCE.Arrival_airport_code → AIRPORT.Airport_code;
9. LEG_INSTANCE.Airplane_id → AIRPLANE.Airplane_id
10. SEAT_RESERVATION.Flight_number → FLIGHT.Flight_number
11. SEAT_RESERVATION.Leg_number → FLIGHT_LEG.Leg_number
12. AIRPLANE.Airplane_type → AIRPLANE_TYPE.Airplane_type_name

Probelem 3

I think the following combinations of columns can be candidate keys

(1) **{Course#}**: Minimal reasonable candidate key. [e.g. ('DSC 461')]

- No duplicated values; No NULL values
- Course# VARCHAR(15)

(2) **{Course#, Univ_Section#}**: This is the most basic and common sense combination. For example, section A of DSC 461 has lectures on TR and section B of DSC 461 has lectures on MWF. [e.g. ('DSC 461', 'A')]

- No duplicated value; No NULL values
- Course# VARCHAR(15)
- Univ_Section# CHAR(5)

(3) **{Course#, Univ_Section#, CreditHours}**: Sometimes, even two courses have the same course number and university section number, they can still be different. For example, the same course offers 3 credits for undergraduate students and 4 credits for graduate students because graduate students are taking this course with an additional project requirment [e.g. ('DSC 461', 'A', 4)]

- No duplicated value; No NULL values
- Course# VARCHAR(15)
- Univ_Section# CHAR(5)
- CreditHours INT(1)

(4) **{Course#, Univ_Section#, Semester}**: Under some circumstances, it is useful to be able to identify even the same course that is offered during different semester. Maybe someone want to figure out why the same course has higher average grade in the summer than during fall/spring semester [e.g. ('DSC 461','A','summer')]

- No duplicated value; No NULL values
- Course# VARCHAR(15)
- Univ_Section# CHAR(5)
- Semester VARCHAR(10) CHECK(Semester IN ('spring','summer','fall','winter','other'))

(5) **{Course#, InstructorName}**: Sometimes, two courses have the same course number but are taught by different instructors due to one for undergraduate students and the other one for graduate students. Or even in some circumstances, the university wants to compare the difference between two same courses taught by different instructors. [e.g. ('DSC 461', 'Eustrat Zhupa')]

- No duplicated value; No NULL values
- Course# VARCHAR(15)
- InstructorName VARCHAR(50)

(6) **{Course#, TimePeriod}**: Sometimes, even with the same course number, two courses are completely different. For example, the course DSC 461 was about Natural Language Processing before 2016. But the instructor of that course left the university and nobody is teaching NLP anymore. So the university makes DSC 461 into a new course about database system but still remains the course number. [e.g. ('DSC 461', 2016)]

- No duplicated value; No NULL values
- Course# VARCHAR(15)
- TimePeriod INT(4)